

PATENT  
Atty. Dkt. No. NVDA/P000721

### **Amendments to the Claims:**

This listing of claims will replace all prior versions and listings of claims in the application:

### **Listing of Claims:**

Claim 1 (Currently Amended): A method for generating vertices of a primitive extension, the primitive extension defining a collection of connected primitives including an originating primitive, comprising:

providing vertices of the originating primitive;

retrieving parameters associated with the primitive extension based on its primitive extension type, the parameters including a width (w), a step size (s) and an anchor width (a); and

generating vertices of the primitive extension using the parameters by: (i) setting the vertices of the originating primitive as the first w vertices of the primitive extension, the vertices of the originating primitive including a number of anchor vertices of the primitive extension, the number of anchor vertices of the primitive extension being equal to the anchor width (a); and (ii) generating one or more additional vertices for each connected primitive other than the originating primitive, the number of vertices generated for said each connected primitive being equal to the step size (s), wherein the primitive extension type includes a triangle fan type, a quadrilateral strip type, and a cube strip type.

Claims 2-3(Cancelled)

Claim 4 (Currently Amended): A method as in claim [[3]] 1, wherein the w, s, a values for the triangle fan type primitive extension are 3, 1, 1, respectively, and the w, s, a values for the quadrilateral strip type primitive extension are 4, 2, 0, respectively, and the w, s, a values for the cube strip type primitive extension are 8, 4, 0, respectively.

**Claim 5 (Previously Presented):** A method as in claim 1, wherein one of the parameters indicates a number of new vertices to be added to form a connected primitive adjacent to the originating primitive.

**Claim 6 (Previously Presented):** A method as in claim 1, wherein one of the parameters indicates a number of the vertices of the originating primitive to be used as anchor vertices for each adjacent connected primitive.

**Claim 7 (Previously Presented):** A method as in claim 1, wherein the parameters indicate a number of the vertices that are shared between two adjacent connected primitives.

**Claims 8-15 (Cancelled)**

**Claim 16 (Currently Amended):** A method for generating a data stream corresponding to a primitive extension, the primitive extension defining a collection of connected primitives including an originating primitive, comprising:  
obtaining vertex data for the originating primitive; and  
retrieving parameters associated with the primitive extension based on its primitive extension type, the parameters including a width (w), a step size (s) and an anchor width (a), and generating an ordered data stream using said parameters;

wherein data elements in the ordered data stream include: (i) vertex data for the originating primitive that includes vertex data for a number of anchor vertex data of the primitive extension, the number of anchor vertex data of the primitive extension being equal to the anchor width (a); and (ii) vertex data for each connected primitive other than the originating primitive, the number of vertex data for said each connected primitive being equal to the step size (s), wherein the primitive extension type includes a triangle fan type, a quadrilateral strip type, and a cube strip type.

**Claims 17-21 (Cancelled)**

**Claim 22 (Previously Presented):** A method as in claim 16, wherein the ordered data stream is generated by a primitive engine, and further comprising providing the ordered data stream to a vertex engine.

**Claim 23 (Previously Presented):** A method as in claim 16, further comprising providing the ordered data stream to a vertex engine.

**Claim 24 (Previously Presented):** A method as in claim 1, wherein the primitive extension parameters are provided through an application program interface (API).

**Claim 25 (Previously Presented):** A method as claim in claim 1, wherein a sufficient number of connected primitives in the primitive extension are generated to approximately cover a surface.

**Claim 26 (Previously Presented):** A method as claimed in claim 1, wherein the vertices of the primitive extension are output as a data stream.

**Claim 27-32 (Cancelled)**

**Claim 33 (Currently Amended):** A method as in claim ~~[[20]]~~ 16, wherein the w, s, a values for the triangle fan type primitive extension are 3, 1, 1, respectively, and the w, s, a values for the quadrilateral strip type primitive extension are 4, 2, 0, respectively, and the w, s, a values for the cube strip type primitive extension are 8, 4, 0, respectively.

**Claim 34 (Currently Amended):** A method for generating vertices of a primitive extension, the primitive extension defining a collection of connected primitives including an originating primitive, comprising:

providing vertices of the originating primitive;

retrieving parameters associated with the primitive extension based on its primitive extension type, the parameters including a width (w), a step size (s) and an anchor width (a); and

generating vertices of the primitive extension using the parameters,  
wherein the primitive extension type includes a triangle fan type, a quadrilateral  
strip type, and a cube strip type.

Claim 35 (Cancelled)

Claim 36 (Previously Presented): A method as in claim 35, wherein the w, s, a values for the triangle fan type primitive extension are 3, 1, 1, respectively, and the w, s, a values for the quadrilateral strip type primitive extension are 4, 2, 0, respectively, and the w, s, a values for the cube strip type primitive extension are 8, 4, 0, respectively.

Claim 37 (Previously Presented): A method as in claim 34, wherein one of the parameters indicates a number of new vertices to be added to form a connected primitive adjacent to the originating primitive.

Claim 38 (Previously Presented): A method as in claim 34, wherein one of the parameters indicates a number of the vertices of the originating primitive to be used as anchor vertices for each adjacent connected primitive.

Claim 39 (Previously Presented): A method as in claim 34, wherein the parameters indicate a number of the vertices that are shared between two adjacent connected primitives.